

Утвержден
РУСБ.10192-01-ЛУ

Н.К. *[Signature]* 9.04.24

СПО РСРД
Руководство системного программиста
РУСБ.10192-01 32 01
Листов 23

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
41543	<i>[Signature]</i> 11.04.24	20310		

АННОТАЦИЯ

Настоящий документ является Руководством системного программиста специального программного обеспечения реализации сервиса распределения данных РУСБ.10192-01 (далее по тексту СПО РСРД).

В документе содержится описание структуры и функционирования, процессы настройки и проверки СПО РСРД.

СОДЕРЖАНИЕ

1. Общие сведения о программе	4
1.1. Назначение программы	4
1.2. Основные функции программы	5
1.3. Требования к техническим и программным средствам	6
2. Структура программы	7
3. Настройка программы.....	10
3.1. Настройка переменных среды	10
3.2. Распространение исполняемых модулей СПО РСРД.....	10
3.3. Конфигурирование компонентов СПО РСРД.....	10
3.4. Конфигурирование компонента реализации сервисов публикации и подписки (DCPS)	11
3.5. Конфигурирование компонента реализации протокола обмена данными (DDSI-RTPS)	13
3.6. Конфигурирование компонента реализации сетевого транспорта данных.....	14
4. Проверка программы	16
4.1. Проверка компилятора объектной модели	16
4.2. Компиляция и выполнение тестовых приложений с библиотеками СПО РСРД	16
4.2.1. Компиляция тестовых приложений.....	16
4.2.2. Выполнение тестовых приложений	17
5. Сообщения системному программисту	19
5.1. Виды (уровни) сообщений СПО РСРД	19
5.2. Текстовый журнал СПО РСРД	20
5.3. Бинарный журнал СПО РСРД	21
Перечень сокращений	22

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1. Назначение программы

СПО РСРД предназначено для использования в качестве инфраструктурной программной подсистемы в территориально распределенных автоматизированных информационно-управляющих системах (АИУС), в том числе, работающих в режиме реального времени. СПО РСРД используется различными приложениями в распределенных вычислительных системах для организации информационного взаимодействия в соответствии с международным стандартом DDS (Data Distribution Service for Real-Time Systems).

Стандарт DDS специфицирует сервис распределения данных, функционирующий по принципу виртуальной глобальной шины данных, к которой могут подключаться участники информационного взаимодействия для модификации или чтения информации, организованной в виде классов объектов и их экземпляров. Для присоединения к общему пространству данных участники взаимодействия используют концепцию публикации и подписки. Публикаторы обеспечивают обновление значений публикуемых данных, а подписчики объявляют состав данных, в получении которых участник заинтересован, а также условия получения таких данных.

Стандарт DDS включает в себя спецификации программного интерфейса, который используется участниками для оперирования данными информационного взаимодействия, и протокол сетевого взаимодействия, обеспечивающий возможность интеграции систем, использующих различные реализации DDS.

DDS обеспечивает высокопроизводительный обмен данными в реальном времени. Распределенные системы, построенные по технологии DDS хорошо масштабируются, благодаря чему DDS может использоваться в системах с низкой пропускной способностью каналов информационного обмена.

СПО РСРД представляет собой отечественную реализацию международного стандарта DDS. Основной областью применения СПО РСРД являются распределенные вычислительные системы, в которых требуется высокопроизводительный обмен данными в реальном времени, в том числе с участием реальных объектов, управляемых человеком (Live-участники в концепции LVC-систем).

СПО РСРД может также применяться для интеграции вычислительных систем во взаимодействующие комплексы, в том числе с участием иностранных систем, использующих другие реализации DDS.

1.2. Основные функции программы

СПО РСРД представляет собой реализацию следующих спецификаций стандарта DDS:

- Data Distribution Services (DDS) версии 1.4 (DCPS (Data Centric Publish-Subscribe) - основной, обязательный уровень стандарта DDS, который описывает модель данных DDS и правила публикации и подписки на данные информационного обмена между участниками);
- Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification (DDSI-RTPS) версии 2.2 (транспортный протокол службы распределения данных);
- ISO/IEC C++ 2003 Language DDS PSM (DDS-PSM-Cxx) (прикладной программный интерфейс DDS для языка программирования C++).

Объем реализации СПО РСРД в основном отвечает следующим профилям соответствия, описанным в спецификациях:

- минимальный профиль (Minimum profile) содержит обязательные к реализации функции DDS;
- профиль управления владением (Ownership profile) — опциональные функции управления владением данными.

В частности, реализованы функции, включающие в себя:

- присоединение к одному или нескольким (до 10) доменам (независимым пространствам данных);
- объявление типов данных (тем), с которыми работает приложение в домене;
- публикацию данных;
- подписку на данные;
- синхронные (статусные условия и списки ожидания) и асинхронные (слушатели) режимы извещения подписчиков о доступности новых данных.

В СПО РСРД реализованы слушатели (Listener) для объектов типа Data Reader и Data Writer.

Таблица 1 описывает характеристики и ограничения СПО РСРД с точки зрения масштабируемости.

Таблица 1 – Характеристики масштабируемости СПО РСРД

Характеристика	Максимальное количество
Количество хостов, на которых выполняются компоненты СПО РСРД	1000

Характеристика	Максимальное количество
Количество одновременно существующих доменов распределения данных	10
Количество участников домена	10000
Количество одновременно используемых при информационном обмене логических сегментов	100
Количество одновременно используемых при информационном обмене тем	1000
Количество публикаторов/подписчиков на одного участника	200
Количество экземпляров объектов на одного публикатора/подписчика	1000000
Количество полей на тему	64

1.3. Требования к техническим и программным средствам

Для функционирования СПО РСРД состав и характеристики технических средств должны соответствовать минимальным требованиям, представленным ниже:

- процессор – Intel 1 ГГц;
- оперативная память – 512 МБ;
- 200 МБ свободного места на диске (для варианта с исходными кодами тестовых федераций и документацией).

Рекомендуемые требования к техническим средствам определяются потребностями прикладных задач (участников), а также размещением участников в вычислительной сети (например, предполагается ли использовать одну ПЭВМ для выполнения нескольких участников). Следовательно, эти параметры должны определяться разработчиками прикладных систем, использующих СПО РСРД.

Для функционирования СПО РСРД на технических средствах должно быть развернуто следующее программное обеспечение:

- операционная система специального назначения «Astra Linux Special Edition» (версия 1.6);
- gcc версии 6.3.0 или старше;
- glibc версии 2.18 или старше;
- qt4-qmake версии 4.8.7 или старше.

2. СТРУКТУРА ПРОГРАММЫ

СПО РСРД включает в себя:

- инструментальные средства разработки объектной модели данных распределенной вычислительной системы;
- заголовочные файлы программного интерфейса DDS для языка программирования C++ (API DDS), необходимые для разработки приложений;
- библиотеки, подключаемые к распределенному приложению и реализующие программный интерфейс приложения к СПО РСРД и функции распространения данных в вычислительной системе.

К инструментальным средствам СПО РСРД относится программное средство компилятор модели данных (ПС КМД). Основная функция ПС КМД - автоматическая генерация классов для прикладных типов данных на языке программирования C++ на основе платформо-независимого описания объектной модели данных. Созданные ПС КМД заголовочные файлы в дальнейшем используются прикладными программами при публикации и подписке на данные, наряду с заголовочными файлами программного интерфейса DDS.

ПС КМД используется разработчиками прикладной распределенной системы каждый раз после внесения изменений в объектную модель данных и перед компиляцией приложений - участников информационного обмена.

Заголовочные файлы программного интерфейса DDS для языка программирования C++ используются разработчиками приложений, предназначенных для работы с СПО РСРД. Они включают в себя копию файлов стандартного программного интерфейса, специфицированного в ISO/IEC C++ 2003 Language DDS PSM и заголовочные файлы реализации стандартного API.

Библиотечные модули СПО РСРД — это модули динамической компоновки, подключаемые к разрабатываемым приложениям и необходимые для их работы. Библиотеки используются как в процессе разработки приложений, так и в эксплуатации прикладных систем, использующих в своей работе СПО РСРД.

СПО включает в себя два библиотечных модуля: `rdds64` и `transport_base`. Оба модуля представлены только в 64-битных вариантах.

Библиотека `rdds64` включает в себя:

- реализацию функций DDS уровня DCPS;
- реализацию проводного протокола DDS - DDSI-RTPS.

Библиотека `transport_base` представляет собой подключаемый модуль базового сетевого транспорта данных.

СПО РСРД не содержит дополнительных компонентов, обеспечивающих функционирование СПО РСРД и приложений. Для получения доступа к сервисам DDS приложению достаточно подключить оба указанных выше библиотечных модуля. Модули СПО РСРД самостоятельно выполняют обнаружение других приложений - участников информационного обмена.

СПО РСРД устанавливается в каталог, указанный пользователем. На него должна ссылаться переменная среды `RDDS_HOME`. Таблица 2 содержит состав файлов этого каталога после установки СПО РСРД.

Таблица 2 – Состав файлов установленного СПО РСРД

Имя файла / каталога	Тип	Назначение
<code>bin</code>	Каталог	Исполняемые файлы: <code>XML_DDS_precompiler</code> — компилятор модели данных; <code>librdds64</code> – библиотека реализации DDS; <code>libtransport_base</code> – библиотека реализации базового сетевого транспорта
<code>bin/resource</code>	Каталог	Ресурсы компилятора модели данных: <code>RDDS_DataTypes.xsd</code> — описание типов данных СПО РСРД
<code>conf</code>	Каталог	Конфигурационные файлы: <code>dcps.cfg</code> – пример конфигурационного файла модуля DCPS; <code>rtps.cfg</code> – пример конфигурационного файла модуля RTPS; <code>transport_base.cfg</code> – пример конфигурационного файла сетевого транспорта
<code>include/isocpp/dds</code>	Каталог	Пользовательские заголовочные файлы API DDS (подключаются в приложениях, использующих СПО РСРД). Структура и состав каталога соответствует спецификациям DDS
<code>include/isocpp/r_dds</code>	Каталог	Внутренние заголовочные файлы реализации API DDS, на которые ссылаются пользовательские заголовочные файлы (не подключаются напрямую в приложениях, использующих СПО РСРД). Структура каталога соответствует структуре <code>include/isocpp/dds</code>
<code>include/isocpp/spec</code>	Каталог	Стандартные заголовочные файлы API DDS, на которые ссылаются пользовательские заголовочные файлы (не подключаются напрямую в приложениях, использующих СПО РСРД). Представляют собой копию спецификаций стандарта

Имя файла / каталога	Тип	Назначение
include/rdds_common	Каталог	Заголовочные файлы, используемые автоматически создаваемыми с помощью компилятора объектной модели файлами приложений: IksException.h, Indent.h, Logger.h, MD5.h, Resource.h, func_def.h, rdds_pack.h, rdds_test_pack.h
resources/strings	Каталог	Строковые ресурсы СПО РСРД: dcps_strings_ru.res, rtps_strings_ru.res, ds_strings_ru.res.
examples/rp/	Каталог	Каталоги и файлы примера для проверки корректности установки СПО РСРД. Включает файл проекта ExampleDemo.pro, файл объектной модели demo_rp.xml, скрипт запуска компилятора объектной модели startCompiler.sh и скрипт запуска теста startExample.sh
examples/rp/demo_rp_DDS	Каталог	Каталог результатов компилирования объектной модели, содержит проект компиляции библиотеки объектной модели demo_rp.pro
examples/rp/publisher	Каталог	Содержит файл проекта приложения публикатора publisher.pro и каталог src с файлом исходного кода приложения publisher.cpp
examples/rp/subscriber	Каталог	Содержит файл проекта приложения подписчика subscriber.pro и каталоги исполняемого модуля публикатора bin, каталог src с файлами исходного кода приложения subscriber.cpp и MainSubscriber.cpp, а также каталог include с заголовочным файлом приложения MainSubscriber.h

3. НАСТРОЙКА ПРОГРАММЫ

3.1. Настройка переменных среды

Перед запуском СПО РСРД должны быть настроены переменные среды, а именно:

- переменная `RDDS_HOME` должна содержать путь установки СПО РСРД;
- переменная `PATH` должна содержать путь к директории `$RDDS_HOME/bin`

Для этого в системной консоли следует выполнить команды:

```
export RDDS_HOME=<Путь установки ПО>
export PATH=$PATH:$RDDS_HOME/bin.
```

Если требуется сделать эти назначения постоянным, команды следует добавить в соответствующий файл настроек (`.bashrc` или ему подобный).

3.2. Распространение исполняемых модулей СПО РСРД

Скомпилированные исполняемые модули СПО РСРД могут быть перенесены на другие хосты с аналогичной конфигурацией системного программного обеспечения. Для этого необходимо выполнить следующие действия:

- 1) скопировать на целевой хост каталог с установленным ПО;
- 2) настроить на целевом хосте переменные среды, как это описано разделе 10.

3.3. Конфигурирование компонентов СПО РСРД

Настройка параметров функционирования компонентов СПО РСРД осуществляется с помощью файлов конфигурации. У каждого компонента существует свой файл конфигурации, а именно:

- файл `dcps.cfg` содержит общие конфигурационные параметры DDS и параметры конфигурации для реализации уровня DCPS спецификаций DDS;
- файл `rtps.cfg` содержит конфигурационные параметры реализации уровня RTPS;
- файл `transport_base.cfg` предназначен для конфигурирования модуля базового транспорта СПО РСРД.

СПО РСРД осуществляет поиск файлов конфигурации в следующей последовательности:

- в текущем каталоге;
- в каталоге `$(RDDS_HOME)/conf`.

Файл конфигурации представляет собой текстовый файл, каждая значимая строка которого должна иметь вид:

<Имя параметра>=<Значение параметра>.

Длина строки не должна превышать 2 Кб. Пустые строки игнорируются. Строка, начинающаяся с символа «#», трактуется как комментарий.

Начало значимой строки до первого вхождения символа равенства трактуется как имя конфигурационного параметра, а оставшаяся часть строки – как его значение. Пробелы в названии конфигурационного параметра и его значении удаляются. Например, следующие строки трактуются одинаково:

«параметр1 = значение»

«параметр1 = значение »

« п а р а м е т р 1 = з н а ч е н и е ».

Имя параметра состоит из имени группы и собственного имени параметра, разделенных знаком «.».

В следующих подразделах описаны параметры каждого файла конфигурации.

3.4. Конфигурирование компонента реализации сервисов публикации и подписки (DCPS)

Функционирование DDS уровня DCPS настраивается как стандартным образом через политики качества обслуживания QoS, выбранные приложением, так и с помощью установки значений параметров конфигурации в файле `dcps.cfg`. Все параметры конфигурации DCPS имеют имя группы «DCPS». Таблица 3 содержит описание этих параметров.

Таблица 3 – Конфигурационные параметры DCPS

Имя параметра	Описание	Возможные значения	Значение по умолчанию
READABLE_LOG_LEVEL	Минимальная важность сообщений, выводимых в текстовый журнал работы СПО РСРД	НЕ ВЫВОДИТЬ БЕЗУСЛОВНОЕ ОШИБКА ПРЕДУПРЕЖДЕНИЕ ИНФОРМАЦИОННОЕ КОНФИГУРАЦИЯ ОТЛАДКА ОТЛАДКА 2	ПРЕДУПРЕЖДЕНИЕ
READABLE_LOG_FILES_NUMBER	Количество файлов текстового журнала, заполняемых циклически	Целое в интервале: 2 — 10	2
READABLE_LOG_FILE_SIZE	Ограничение на размер файла текстового журнала (байт)	Целое в интервале: 100000 — 2000000000	1000000

Имя параметра	Описание	Возможные значения	Значение по умолчанию
BINARY_LOG_LEVEL	Минимальная важность сообщений, выводимых в бинарный (сжатый) журнал работы СПО РСРД	НЕ_ВЫВОДИТЬ БЕЗУСЛОВНОЕ ОШИБКА ПРЕДУПРЕЖДЕНИЕ ИНФОРМАЦИОННОЕ КОНФИГУРАЦИЯ ОТЛАДКА ОТЛАДКА 2	КОНФИГУРАЦИЯ
BINARY_LOG_FILES_NUMBER	Количество файлов бинарного журнала, заполняемых циклически	Целое в интервале: 2 — 10	10
BINARY_LOG_FILE_SIZE	Ограничение на размер файла бинарного журнала (байт)	Целое в интервале: 100000 — 2000000000	1000000
MAX_SAMPLES_PER_DW_LIMIT	Ограничение ресурсов на количество сэмплов, обрабатываемых каждым DataWriter	Целое в интервале: 1 — 10000	1024
MAX_SAMPLES_PER_DR_LIMIT	Ограничение ресурсов на количество сэмплов, обрабатываемых каждым DataReader	Целое в интервале: 1 — 10000	1024
NUMBER_OF_PREALLOCATED_SAMPLES_PER_READER	Количество сэмплов, размещенных статически на DataReader	Целое в интервале: 0 — 1000	10
COLLECT_GARBAGE_PERIOD	Период сборки мусора в секундах (0 - не производится)	Целое в интервале: 0 — 3600	300

Параметры журнала работы определяют, какие сообщения должны выводиться в журнал (посредством задания минимального уровня выводимых сообщений), а также характеристики файлов журнала.

Поддерживается два вида журналов работы: текстовый и бинарный. Текстовый журнал работы удобен тем, что его может прочитать пользователь без применения дополнительных инструментов. Но текстовый журнал часто имеет большой размер, что делает его неудобным для вывода детальной и отладочной информации о работе DDS. Для таких целей поддерживается бинарный журнал работы. Для его чтения необходимы специальные утилиты, но он удобен для вывода отладочной информации с последующей передачей журнала работы разработчикам DDS.

Фрагменты журналов содержатся в отдельных файлах, количество которых задается параметром `<>_LOG_FILES_NUMBER`. Параметр `<>_LOG_FILE_SIZE` задает размер одного фрагмента. При достижении максимального количества фрагментов

запись снова осуществляется в первый фрагмент, и далее фрагменты перезаписываются по циклу.

Параметры `MAX_SAMPLES_PER_DW_LIMIT` и `MAX_SAMPLES_PER_DR_LIMIT` задают максимальное количество сэмплов, которые может хранить каждый объект типа `DataWriter` и `DataReader` соответственно. Если приложение посредством политик качества обслуживания задает большее значение или указывает «без ограничений», то устанавливается именно это значение из файла конфигурации.

Параметр `NUMBER_OF_PREALLOCATED_SAMPLES_PER_READER` позволяет оптимизировать выделение памяти для ресурсов объекта типа `DataReader`. Для указанного в параметре количества экземпляров (сэмплов) данных память предразмещается при инициализации, и лишь при отсутствии свободных слотов в предразмещенной области память для сэмпла выделяется динамически при его получении. Это может быть полезно в основном для структур данных фиксированного размера, не содержащих внутри себя полей, требующих динамического размещения.

Параметр `COLLECT_GARBAGE_PERIOD` управляет работой встроенного сборщика мусора.

3.5. Конфигурирование компонента реализации протокола обмена данными (DDSI-RTPS)

Конфигурационные параметры компонента, содержащиеся в файле `rtps.cfg`, включают в себя данные настройки функционирования протокола RTPS и значения технологических параметров, влияющих на работу самого компонента. Имя группы для всех параметров конфигурации компонента реализации протокола обмена данными — `BUSINESS`.

Таблица 4 описывает значения этих конфигурационных параметров.

Таблица 4 – Конфигурационные параметры компонента реализации протокола обмена данными

Имя параметра	Описание	Возможные значения	Значение по умолчанию
<code>DEFAULT_UNICAST_IP</code>	Индивидуальный IP адрес, используемый для сетевого обмена данными по умолчанию	Любой действительный unicast IP адрес	127.0.0.1
<code>DEFAULT_MULTICAST_IP</code>	Групповой IP адрес, используемый для сетевого обмена пользовательскими данными по умолчанию	Multicast IP адрес из диапазона 239.0.0.0/8	239.0.0.2
<code>HEARTBEAT_PERIOD</code>	Период рассылки анонсов в случае stateful алгоритма, мс	Целое в интервале: 100 - <code>MAXINT</code>	10000

Имя параметра	Описание	Возможные значения	Значение по умолчанию
HEARTBEAT_RESPONSE_DELAY	Задержка отправки подтверждения/запроса в случае stateful алгоритма, мс	Целое в интервале: 0 - MAXINT	500
NACK_RESPONSE_DELAY	Задержка отправки данных в ответ на запрос в случае stateful алгоритма, мс	Целое в интервале: 0 - MAXINT	200
NACK_SUPPRESSION_DURATION	Период подавления последовательных запросов, мс	Целое в интервале: 0 - MAXINT	0
PUSH_MODE	Признак включения режима безусловной отправки данных	ДА НЕТ	ДА
RESEND_PERIOD	Период рассылки данных в случае stateless алгоритма, мс	Целое в интервале: 100 - MAXINT	30000
CHECK_INCOMING_PERIOD	Период автопроверки входящих сообщений, мс	Целое в интервале: 1000 - MAXINT	10000

3.6. Конфигурирование компонента реализации сетевого транспорта данных

Конфигурационные параметры реализации сетевого транспорта данных содержат значения технологических параметров, влияющих на работу данного компонента. Имя группы для всех параметров конфигурации компонента — TRANSPORT.

Таблица 5 описывает значения этих конфигурационных параметров.

Таблица 5 – Конфигурационные параметры компонента реализации сетевого транспорта данных

Имя параметра	Описание	Возможные значения	Значение по умолчанию
NUMBER_OF_BUFFERS	Количество буферов для приема данных (делятся между всеми приемниками)	Целое в интервале: 1 - MAXINT	1024
BUFFER_SIZE	Размер одного буфера для приема данных, байт	Целое в интервале: 1 - 65507	65507
REPORT_TIMEOUT	Период автопроверки нити уведомления о получении данных, сек	Целое в интервале: 1 - 60	5
DISPATCHER_TIMEOUT	Период автопроверки нити управления передачей данных, сек	Целое в интервале: 1 - 60	5
SELECT_TIMEOUT	Период автопроверки нити приема данных, сек	Целое в интервале: 1 - 60	5

Имя параметра	Описание	Возможные значения	Значение по умолчанию
MULTICAST_INTERFACE	IP адрес сетевого интерфейса, через который будет производиться многоадресный обмен данными. При значении данного параметра 0.0.0.0 приём данных будет производиться через все доступные интерфейсы, а передача — через интерфейс по выбору операционной системы	IP адрес любого сетевого интерфейса хоста или 0.0.0.0	0.0.0.0
MAX_TR_DATA_SIZE	Максимальный размер передаваемого блока данных, байт. Не включает в себя размер заголовка низлежащего протокола	Целое в интервале: 1000 - 65507	65507

3.7. Настройка ПС КМД

Настройки ПС КМД не требуется.

4. ПРОВЕРКА ПРОГРАММЫ

4.1. Проверка ПС КМД

Проверка ПС КМД производится с использованием примера, приведенного в каталоге `examples/rp`. Перед проверкой ПС КМД необходимо убедиться, что каталог `examples/rp/demo_rp_DDS` содержит только один файл проекта `demo_rp.pro`, соответственно, остальные каталоги или файлы должны быть удалены. После этого в каталоге `examples/rp` запускается скрипт `startCompiler.sh`, который запускает компилятор объектной модели для обработки объектной модели примера `demo_rp.xml`. ПС КМД установлен правильно, если после его работы в каталоге `demo_rp_DDS/include` появляются четыре файла: `demo_rp.h`, `demo_rp_DCPS.h`, `demo_rp_DCPS.hpp` и `demo_rp_DCPS_Pack.h`, а в каталоге `demo_rp_DDS/scr` два файла: `demo_rp.cpp` и `demo_rp_DCPS_Pack.cpp`.

4.2. Компиляция и выполнение тестовых приложений с библиотеками СПО РСРД

4.2.1. Компиляция тестовых приложений

Для компиляции тестовых приложений необходимо создать классы для прикладных типов данных тестового приложения. После этого создаются библиотека прикладных типов данных и приложения отправителя и получателя.

Для создания библиотеки прикладных типов вначале используется утилита `qmake-qt4`, которая обрабатывает файл проекта `demo_rp.pro` и создает необходимый файл компиляции и сборки библиотеки. Для этого необходимо в каталоге `examples/rp/demo_rp_DDS` выполнить команду:

```
qmake-qt4 demo_rp.pro.
```

В результате работы утилиты в текущем каталоге появится каталог `lib` и файл компиляции проекта `Makefile`. Для компиляции собственно библиотеки в текущем каталоге надо вызвать утилиту `make`, которая создаст в каталоге `lib` библиотеку с именем `libdemo_rp.a`.

Далее в любой последовательности нужно скомпилировать и собрать приложения. Для этого запускается по очереди утилита `qmake-qt4` в каталогах `publisher` и `subscriber` с соответствующим параметром: `publisher.pro` или `subscriber.pro`, после чего выполняется в каждом каталоге утилита `make`.

Если компиляция и сборка приложений выполнялась без ошибок, то в каталогах `publisher/bin` и `subscriber/bin` должны появиться исполняемые файлы `publisher` и `subscriber` соответственно.

4.2.2. Выполнение тестовых приложений

Перед запуском приложений необходимо убедиться, что конфигурационные параметры (см. Таблицы 3-5) настроены правильно. Для запуска приложений локально можно воспользоваться скриптом `startExample.sh`. Данный скрипт в отдельном окне терминала с именем `pub` запускает приложение публикатор, а в текущем окне запускает приложение подписчик.

Приложение публикатор работает по следующему алгоритму:

- создание участника RDDS и соответствующих топиков данных;
- ожидание команды «Старт»;
- отправка пяти сэмплов с частотой 1 Гц;
- ожидание команды «Стоп»;
- завершение работы приложения.

На терминале публикатора после завершения работы должна быть видна следующая информация:

```
*** QoS BestEffort is ON. History=5. ***
*** QoS Shared is ON... ***
*** QoS Deadline is INFINITE ***
*** QoS LIVESPAN is OFF ***
*** QoS DURABILITY = VOLATILE ***
Waiting for command Start...
Start...
Send sample 0 number=10
Send sample 1 number=11
Send sample 2 number=12
Send sample 3 number=13
Send sample 4 number=14
All data have been sent ...
Finish...
```

Приложение подписчик ждет появления публикатора, шлет команду «Старт», принимает сэмплы данных и после их получения шлет команду «Стоп», а затем завершает работу. В окне терминала подписчика при корректном завершении работы будет отображаться следующая информация:

```
Test : publisher : :: sub :  
*** QoS BestEffort is ON. History=6. ***  
*** QoS Shared is ON... ***  
*** QoS DURABILITY = VOLATILE ***  
*** QoS Deadline is INFINITE ***  
Waiting for writers ...  
Sending 'Start' command ...  
Waiting for samples...  
Received sample 0 from DataWriter with id=1 number 10 Time=XXX  
Received sample 1 from DataWriter with id=1 number 11 Time=XXX+1  
Received sample 2 from DataWriter with id=1 number 12 Time=XXX+2  
Received sample 3 from DataWriter with id=1 number 13 Time=XXX+3  
Received sample 4 from DataWriter with id=1 number 14 Time=XXX+4  
Quantity of received samples = 5  
Sending 'Stop' command ...  
Finish...,
```

где XXX – текущее время в секундах.

Для запуска приложений на разных хостах необходимо соответствующим образом настроить конфигурационные параметры и в командной строке терминалов запускать соответствующее приложение. Сообщения приложений при корректной их работе не меняются.

5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

5.1. Виды (уровни) сообщений СПО РСРД

СПО РСРД может выводить сообщения в журнал в соответствии с заданным в конфигурационных файлах минимальным уровнем вывода сообщений. Уровни характеризуют степень значимости (важности) сообщений. Параметр `DCPS.READABLE_LOG_LEVEL` определяет, что в журнал должны выводиться все сообщения, уровень которых равен заданному значению или выше.

Минимальный уровень вывода сообщений в файл журнала может устанавливаться в следующих значениях (от более значимых к менее значимым):

- НЕ_ВЫВОДИТЬ — при установке данного минимального уровня вывода сообщений СПО РСРД не выводит никаких сообщений в файл журнала и не создает даже пустого файла (и, следовательно, не обращается к диску по записи). Данный уровень введен для обеспечения возможности СПО РСРД функционировать на устройствах без возможности изменять данные на их дисках;
- БЕЗУСЛОВНОЕ — сообщения с таким уровнем вывода выводятся всегда, независимо от значения конфигурационных параметров (за исключением случая, когда задано значение «НЕ_ВЫВОДИТЬ»);
- ОШИБКА — сообщение информирует о серьезной ошибке, возникшей при выполнении компонента СПО РСРД, которая либо блокирует выполнение СПО РСРД (и приложение завершается), либо (при продолжении выполнения приложения) приводит к тому, что выполнение СПО РСРД своих функций не гарантируется;
- ПРЕДУПРЕЖДЕНИЕ — сообщение информирует о подозрительной ситуации, не мешающей выполнению компонента СПО РСРД;
- ИНФОРМАЦИОННОЕ — любое сообщение, не имеющее отношения к сбоям или подозрительным ситуациям, например, вывод статистики выполнения СПО РСРД;
- КОНФИГУРАЦИЯ — используется для сообщений, информирующих о текущей конфигурации СПО РСРД;
- ОТЛАДКА — используется для вывода отладочной информации;
- ОТЛАДКА_2 — используется для вывода детальной отладочной информации.

Следует иметь в виду, что вывод в журнал – ресурсоемкая операция, поэтому не рекомендуется задавать низкий уровень вывода сообщений при реальной работе СПО РСРД.

Поддерживается два вида журналов работы: текстовый и бинарный. Журналы организованы как циклический набор файлов. Количество и размер файлов задаются конфигурационными параметрами `READABLE_LOG_FILES_NUMBER` и `READABLE_LOG_FILE_SIZE` для текстового журнала и `BINARY_LOG_FILES_NUMBER` и `BINARY_LOG_FILE_SIZE` для бинарного журнала.

5.2. Текстовый журнал СПО РСРД

Текстовый журнал СПО РСРД предназначен для «быстрого» анализа работы приложений, использующих СПО РСРД. По умолчанию уровень вывода в текстовый журнал устанавливается в ПРЕДУПРЕЖДЕНИЕ. Таким образом, текстовый журнал будет содержать информацию только об ошибках выполнения и подозрительных ситуациях, которые могут вызвать сбои в работе приложений.

Текстовый журнал представляет собой набор записей в кодировке UTF-8. Кодировка не зависит от платформы, на которой выполняется приложение. Записи в файле имеют следующую структуру:

«ХОСТ ДАТА ВАЖНОСТЬ ИСТОЧНИК СООБЩЕНИЕ»,

где

- ХОСТ – IP-адрес хост-системы;
- ДАТА – дата и время вывода сообщения;
- ВАЖНОСТЬ – уровень значимости сообщения;
- ИСТОЧНИК – компонент СПО РСРД, сформировавший сообщение;
- СООБЩЕНИЕ – собственно текст сообщения (возможно, многострочный).

Формат вывода в журнал даты, следующий:

ГГГГ.ММ.ДД ЧЧ:ММ:СС.ССС

где

- ГГГГ — четыре цифры номера года;
- ММ — две цифры номера месяца. Если номер месяца меньше 10, то первая цифра устанавливается равной «0»;
- ДД — две цифры номер дня в месяце. Если номер дня в месяце меньше 10, то первая цифра устанавливается равной «0»;
- ЧЧ — две цифры часа в 24-часовом формате. Если номер часа меньше 10, то первая цифра устанавливается равной «0»;

- ММ — две цифры минут. Если номер минуты меньше 10, то первая цифра устанавливается равной «0»;
- СС — две цифры секунд. Если номер секунды меньше 10, то первая цифра устанавливается равной «0»;
- ССС — три цифры миллисекунд. Если номер миллисекунды не трехзначный, то одна или две первые цифры устанавливаются равными «0».

Например, сообщение об ошибке при приеме данных будет записано следующим образом:

```
127.000.001.001      2016.05.18      11:58:10.289      ПРЕДУПРЕЖДЕНИЕ
RTPSCommunicationObject::readMsg  Ошибка чтения сэмпла RTPS N:
100000000 - Не найден сэмпл по предоставленным данным. Выполнение
продолжено.
```

5.3. Бинарный журнал СПО РСРД

Бинарный журнал СПО РСРД предназначен для отладки сложных приложений и поиска возможных причин неправильного функционирования. Предполагается, что бинарный журнал будет использоваться специалистами службы поддержки пользователей поставщика СПО РСРД. По умолчанию уровень вывода в бинарный журнал устанавливается в КОНФИГУРАЦИЯ. Таким образом, бинарный журнал будет содержать, кроме сообщений об ошибках выполнения и подозрительных ситуациях, также и расширенную информацию о функционировании СПО РСРД и установленные при выполнении приложения настройки конфигурации.

Бинарный журнал представляет собой набор записей. Формат бинарного журнала сделан более компактным, что упрощает передачу файлов журналов в службу поддержки. Для этого вместо текстового описания сообщения в журнал записывается только код сообщения и значения переменных.

Непосредственная работа с бинарным журналом невозможна. Специалисты службы поддержки после получения файлов бинарного журнала переводят его в читаемый вид для анализа ситуации.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС	- операционная система
ПО	- программное обеспечение
ПС КМД	- программное средство компилятор модели данных
ПЭВМ	- персональная электронно-вычислительная машина
РСРД	- реализация сервиса распределения данных
СПО	- специальное программное обеспечение
API	- Application Program Interface (программный интерфейс приложения)
DCPS	- Data Centric Publish-Subscribe (компонент реализации сервисов подписки и публикации)
DDS	- Data Distribution Service (сервис распределения данных)
DDSI-	- DDS Interoperability Real-time Publish-Subscribe Wire Protocol (транспортный
RTPS	протокол службы распределения данных)
LVC	- Live-Virtual-Constructive (живое-виртуальное-сконструированное)
PSM	- Platform Specific Model (прикладной программный интерфейс)
QoS	- Quality of Service (политика качества обслуживания)
RTPS	- Real-time Publish-Subscribe (компонент транспортного протокола службы
XML	- eXtensible Markup Language (расширяемый язык разметки)
XSD	- XML Schema (схема XML документа)

[illegible]

Изм.

Номера листов (страниц)

ИЗМЕН-
ЕННЫХ

замененных

НОВЫХ

анну-
лирован-
ных

Всего
листов
(страниц)
в докум.

№
документа

Входящий
№ сопроводительного
документа и
дата

Подп.

Дата	Время	События
12.01.2025	08:00	Получение задания на разработку нового функционала.
12.01.2025	10:00	Совещание с командой для обсуждения требований.
12.01.2025	14:00	Начало работы над задачей.
12.01.2025	18:00	Завершение работы на сегодня.
13.01.2025	09:00	Продолжение разработки.
13.01.2025	12:00	Обед.
13.01.2025	13:00	Продолжение разработки.
13.01.2025	17:00	Тестирование разработанных модулей.
13.01.2025	19:00	Подготовка отчета по результатам работы.
14.01.2025	08:30	Согласование отчета с руководителем.
14.01.2025	10:30	Внесение изменений в план работ.
14.01.2025	14:30	Продолжение разработки.
14.01.2025	18:30	Завершение работы на сегодня.

3

BCE

РД СБ. 14.02-24
от 03.01.2024



11.04.24